



# Traitement des valences verbales dans une grammaire catégorielle du français

Karine Kray-Baschung, Gabriel G. Bès, Martine Emorine

## ► To cite this version:

Karine Kray-Baschung, Gabriel G. Bès, Martine Emorine. Traitement des valences verbales dans une grammaire catégorielle du français. Informatique et Langue naturelle, 1989, Nantes, France. pp.269-283. halshs-00372930

**HAL Id: halshs-00372930**

**<https://shs.hal.science/halshs-00372930>**

Submitted on 7 Aug 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## TRAITEMENT DES VALENCES VERBALES DANS UNE GRAMMAIRE CATEGORIELLE DU FRANCAIS <sup>1</sup>

Karine Baschung, Gabriel G. Bès, Martine Emorine

Université Blaise Pascal Clermont II  
Formation Doctorale Linguistique et Informatique  
34 avenue Carnot  
63 037 Clermont-Ferrand Cedex  
France  
{karine, gabriel, martine}@ucfsl.uucp

### 0. INTRODUCTION

Dans une grammaire catégorielle classique (e.g. Bar-Hillel (1953)), la notion de *catégorie* sert à représenter les constituants ainsi que l'ordre linéaire dans lequel ces derniers sont susceptibles d'apparaître. Le traitement de la linéarité s'effectue au moyen (a) de l'ordre des valences dans une liste et (b) de l'utilisation du 'slash' directionnel signalant que l'argument doit être trouvé à droite ou à gauche du foncteur.

En UCG (*Unification Categorical Grammar*, cf. Zeevat, Klein & Calder (1987)), l'ordre des mots est traité de même, à la différence près que le 'slash' directionnel se trouve remplacé par un trait binaire *Ordre* dont les valeurs sont *pre* ou *post*. La catégorie S/NP\NP d'une grammaire catégorielle normale est donc représentée comme S/NP:post/NP:pre en UCG où *pre* et *post* indiquent respectivement que le foncteur doit précéder ou bien suivre son argument.

Il est possible selon nous de rendre compte dans le cadre d'UCG des phénomènes complexes de la linéarité en français, moyennant deux aménagements : nous postulons en effet que les valences verbales ne sont pas associées a priori à une valeur quelconque pour l'ordre, et qu'elles font partie d'un ensemble plutôt que d'une liste. Le comportement syntaxique des constituants en surface est donc dissocié de l'ordre des valences et régulé par un système de traits similaire à celui de Karttunen (1986) ou d'Uszkoreit (1987).

### 1. UCG

En UCG, l'information relative à la phonologie, à la catégorie syntaxique, à la sémantique et à l'ordre d'un mot constitue une entité unique de la grammaire appelée *signe*. Un signe peut se représenter horizontalement ou verticalement, comme en (1), et ses différents champs sont séparés par un double point:

---

<sup>1</sup> Les recherches présentées dans cet article ont été partiellement conduites dans le cadre du Projet ESPRIT 393 ACORD *The Construction and Interrogation of Knowledge Bases using Natural Language Text and Graphics* de la CEE.

(1) Signe UCG

Phonologie: Catégorie: Sémantique: Ordre

Phonologie:  
Catégorie:  
Sémantique:  
Ordre

Nous laisserons de côté l'information relative à la sémantique et à la phonologie, pour nous intéresser principalement aux champs *Catégorie* et *Ordre*. Suivant la tradition catégorielle, les catégories d'UCG peuvent être basiques ou complexes. Les catégories de base ont la forme *Tête^Traits* où la *Tête* est l'un des symboles atomiques *n(oun)*, *n(oun)p(hrase)* ou *s(entence)*, et où les *Traits* sont une liste de traits. Les catégories complexes ont la forme *C/Signe*, où *C* est atomique ou bien complexe et où *Signe* est un signe (en conséquence de quoi un foncteur impose des contraintes sur le signe argument en entier et non pas seulement sur sa catégorie, contrairement à ce qui se passe dans une grammaire catégorielle classique). On appelle *partie active* d'une catégorie complexe la partie dépourvue de la spécification *Tête^traits*. Le premier *signe accessible* de la partie active est appelé *signe actif*; chaque fois qu'une valence verbale est consommée, c'est donc le signe suivant de la partie active qui devient le nouveau signe actif. Le champ *Ordre* conditionne l'application des règles de combinaison: l'information *pre* dans un signe argument *Y* indique que le foncteur *X/Y* doit précéder l'argument, tandis que l'information *post* indique que le foncteur doit suivre l'argument. En UCG, la règle d'application fonctionnelle en avant <sup>2</sup> a la forme suivante, où il est fait usage de termes et d'unification entre termes, et où les majuscules indiquent des variables Prolog:

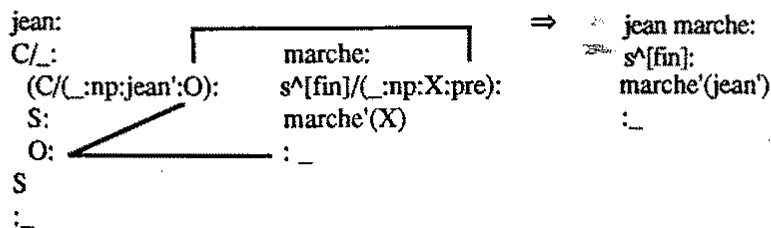
(2) Application en avant (FA)

Foncteur	Argument	⇒	Résultat
PhonologieF:			PhonologieF PhonologieA:
CatégorieF/PhonologieA:	PhonologieA:		CatégorieF:
CatégorieA:	CatégorieA:		SémantiqueF:
SémantiqueA:	SémantiqueA:		OrdreF
pre:	pre		
SémantiqueF:			
OrdreF			

La règle exige que la partie active du signe foncteur unifie avec le signe argument. Le *Résultat* est un signe identique au signe foncteur, sauf que la partie active de la catégorie complexe disparaît et que les variables partagées par la partie active du foncteur et le reste du signe foncteur sont susceptibles d'avoir été instanciées par l'unification de la partie active et de l'argument. La phonologie du résultat est la concaténation de la phonologie du foncteur et de celle de l'argument. Soit à titre d'exemple la dérivation de la phrase *Jean marche*:

<sup>2</sup> L'application fonctionnelle en arrière constitue le symétrique rigoureux de (2), sauf que l'argument précède le foncteur et que *pre* est remplacé par *post*.

(3) Dérivation de *Jean marche*



En (3), les lignes matérialisent le flux de l'information déterminant pour l'ordre: les variables partagées font en sorte que l'information *pre* qui se trouve dans une valence verbale contraigne le foncteur NP (au type monté) qui consomme cette même valence à précéder le verbe.

## 2. LES DONNEES LINGUISTIQUES

L'ordre semi-libre du français est caractérisable par trois observations centrales. Premièrement, la positionnement à droite ou à gauche du verbe est relativement libre pour un argument donné. En effet, en fonction de sa classe morphologique et du type de la phrase (déclarative, inversion sujet-auxiliaire, question-wh, etc.), le SN sujet peut apparaître à la gauche ((4a)) ou à la droite ((4b,c)) du verbe, ou bien entre l'auxiliaire et le verbe ((4d)):

- (4) (a) *Jacques aime Marie.*  
 (b) *Aime-t-il Marie ?*  
 (c) *Quel livre aime Jacques ?*  
 (d) *A-t-il aimé Marie ?*

Les autres arguments du verbe sont libres, sous des conditions similaires, d'apparaître d'un côté ou de l'autre du verbe. Par exemple, un SN non-nominatif ne peut jamais s'insérer à la gauche d'un verbe, mais les clitiques et les constituants interrogatifs le peuvent:

- (5) (a) *\*Marie a regardée Jacques ? (avec Marie = Objet)*  
 (b) *Quelle revue a regardée Jacques ? (avec Quelle revue = Objet)*  
 (c) *Jacques l'a regardée.*

Deuxièmement, il ne semble pas exister de régularités claires en ce qui concerne l'ordre relatif dans une séquence d'arguments. Autrement dit, si l'on admet seulement la combinaison de constituants adjacents, et si l'on fait l'hypothèse que les combinaisons s'effectuent de gauche à droite, on aboutit par exemple aux variations de (6), où les distributions autorisées sont encore une fois reliées à des facteurs tels que la classe morphologique des constituants et le mode du verbe:

- (6) (a) *Pierre donne à Marie un livre.* [Sujet, Objet Indirect, Objet]  
 (b) *Pierre donne un livre à Marie.* [Sujet, Objet, Objet Indirect]  
 (c) *Le lui donne-t-il ?* [Objet Indirect, Objet, Sujet]  
 (d) *Se le donne-t-il ?* [Objet, Objet Indirect, Sujet]

Troisièmement, il y a des restrictions de cooccurrence entre constituants. Par exemple, les clitiques peuvent conditionner le positionnement et la classe d'autres arguments: <sup>3</sup>

- (7) (a) Pierre le *lui* donne  
(b) Pierre *lui* en donne  
(c) Pierre *lui* donne un livre  
(d) \*Pierre *lui* le donne  
(e) \*Pierre *lui* y donne

Etant donné que l'ordre et le positionnement des arguments du verbe sont très flexibles en français, l'ordre rigide introduit en UCG dans la liste active ainsi que le placement fixe résultant des valeurs d'*Ordre* paraissent inadéquats. Mais il n'est pas vrai non plus par ailleurs que l'ordre en français soit tout à fait libre; de fait, il semble plutôt régi par des principes conditionnels du type suivant:

- (8) SI (a) le verbe possède une valence objet, et  
(b) le SN objet est un constituant interrogatif, et  
(c) le verbe se trouve à une forme fléchie simple, et  
(d) le clitique t-il/t-elle n'a pas été incorporé

ALORS l'objet peut se placer à la gauche ou à la droite du verbe

Mettons que la condition (8d) ne soit pas satisfaite: le SN interrogatif peut alors se placer à gauche uniquement, comme en témoigne la phrase \**Jacques a-t-il regardé quelle revue?* - et il en va de même, mutatis mutandis, pour les autres conditions. Sur un plan général, il est possible de distinguer cinq critères dont l'interaction détermine si un argument donné est susceptible d'occuper ou non une position donnée dans une phrase:

- (9) (a) Position (gauche ou droite) par rapport au verbe,  
(b) Forme du verbe et type de la phrase,  
(c) Classe morphologique (lexicale, clitique ou interrogative) du dernier élément à avoir concaténé à gauche ou à droite du verbe,  
(d) Classe morphologique (lexicale, clitique ou interrogative) du constituant en question,  
(e) Cas.

Dans la suite de cet article nous montrons comment il est possible d'étendre UCG de manière à exprimer l'ensemble de ces données linguistiques. La grammaire résultante est capable d'en rendre compte sans avoir recours à l'ambiguïté lexicale ou à des règles de permutation. <sup>4</sup>

### 3. EXTENSIONS A UCG

Les deux modifications radicales introduites par nous à UCG sont les suivantes. Premièrement, la partie active d'une catégorie verbale se trouve représentée comme un ensemble plutôt que comme une liste; deuxièmement, un système de traits est destiné à réguler les interactions entre les critères sur l'ordre décrits en (9).

<sup>3</sup> Les items en italiques sont ceux dont les restrictions de cooccurrence sont violées (phrases agrammaticales) ou respectées (phrases grammaticales); e.g. (7d) est agrammaticale car *lui* ne peut pas être suivi par *le*.

<sup>4</sup> Une règle de permutation a par exemple la forme  $X/Y, Y/Z \Rightarrow X/Z$ , où  $X/Y$  représente un SN au type monté, et  $Y/Z$  un verbe argument (cf. Baschung et al. (1987)).

### 3.1. STRUCTURE DU SIGNE ET REGLE DE COMBINAISON : DE LA LISTE ACTIVE A L'ENSEMBLE ACTIF

La nouvelle grammaire FG (*French Grammar*, cf. Bès & Gardent (1989), Gardent, Bès, Jurie & Baschung (1989) et Baschung (1990)) modifie comme suit la structure du signe:

#### (10) Signe FG

Phonologie:  
Catégorie:  
Valences:  
Traits:  
Sensibilité aux Règles:  
Sémantique:  
Optionalité

Les champs *Sémantique* et *Phonologie* fonctionnent exactement comme en UCG standard. Il en va quasi de même pour le champ *Catégorie*, à ceci près qu'en FG (i) il n'existe pas de traits associés à la tête, et que (ii) la liste des valences -traitée comme un ensemble (cf. § 3.3)- constitue un champ à part entière. Il y a un champ *Optionalité* indiquant si le constituant est obligatoire ou non, mais pas de champ spécial pour l'ordre. Les traits sont accessibles indépendamment de la catégorie et regroupés dans les champs *Traits* et/ou *Sensibilité aux Règles*. Dans ce qui suit, nous ne faisons mention que des traits directement liés aux contraintes d'ordre, à savoir le cas (représenté par la variable *Cas*), la forme du verbe (*FormeV*), la classe morphologique (*Classe*) des syntagmes nominaux -i.e. lexicale (*lex*), clitique (*le*, *lui*, ...) ou interrogative (*wh*)- ou des verbes, ainsi que la classe du dernier constituant à avoir concaténé à gauche (*ImF*) ou à droite (*ImB*). Cette dernière information est remise à jour à chaque application de la règle de combinaison (cf. (13)); le signe associé à l'énoncé *Jean lui a donné un livre* possédant par exemple la valeur *lex* pour *ImF* et *ImB* tandis que *lui a donné un livre* possède *lui* et *lex* respectivement. On trouve en (11) une représentation simplifiée du champ *Traits*, où l'on voit qu'un même trait est susceptible d'occuper une position différente suivant le type des entités linguistiques; e.g. le champ *Traits* des valences verbales et du signe actif des SN (traits de valence) vs. le champ *Traits* des signes verbaux et des SN (traits de signe):

- |      |     |   |  |
|------|-----|---|--|
| (11) | (a) | <u>Traits de valence du champ <i>Traits</i></u> | [ <i>Cas</i> , <i>Classe</i> , <i>ImF</i> , <i>ImB</i> ]       |
|      | (b) | <u>Traits de signe du champ <i>Traits</i></u>   | [ <i>Classe</i> , ( <i>ImF</i> : <i>ImB</i> ), <i>FormeV</i> ] |

Le champ *Sensibilité aux Règles*, représenté en (12), comporte une série de traits qui peuvent être différents selon que l'on combine en avant (*fc*, pour *forward combination*) ou en arrière (*bc*, pour *backward combination*), c'est pourquoi il est divisé en deux:

#### (12) Traits de signe dans le champ *Sensibilité aux Règles* d'un signe actif

ConditionsF  $\Rightarrow$  *fc*  $\Rightarrow$  *Traits\_résultantF*, ConditionsB  $\Rightarrow$  *bc*  $\Rightarrow$  *Traits\_résultantB*

*Conditions* et *Traits\_résultant* sont des listes de traits dont l'ordre et les valeurs sont indépendants de ceux du champ *Traits*. L'idée intuitive sous-jacente à cette répartition est que les foncteurs (e.g. les SN au type monté) sont de nature fondamentalement duelle, i.e. qu'ils concatènent à droite ou à gauche sous des conditions différentes et avec des résultats différents. Via l'unification entre variables dans la règle de combinaison, les traits *Conditions* imposent des contraintes sur les traits de l'argument, tandis que les *Traits\_résultant* sont hérités par le signe résultat. En omettant les champs *Sémantique* et *Optionalité*, la règle de combinaison en avant a l'allure suivante:

(13) Combinaison en avant (FC)

<i>Foncteur</i>	<i>Argument</i>
PhonologieF:	PhonologieA:
CatégorieF/PhonologieA:	CatégorieA:
ValencesA:	ValencesA':
[ClasseA, ...]	[ClasseA, (ImF: ImB), FormeV]
[(ImF: ImB), FormeV] $\Rightarrow$ fc $\Rightarrow$	:-
[(ClasseF: _), FormeV'], _)	
:-: _:-	
(combine(ValencesA, ValencesA', ValencesR))	
$\Rightarrow$ <i>Résultat</i>	
PhonologieF PhonologieA:	
CatégorieF:	
ValencesR:	
[ClasseA, (ClasseF: ImB), FormeV]	
:-	

La catégorie du résultat est la catégorie du foncteur (*CatégorieF*). La règle exige que les valences du signe actif du foncteur (*ValencesA*) et les valences de l'argument (*ValencesA'*) se combinent pour constituer les valences du résultat (*ValencesR*). La notion de combinaison repose sur l'idée que les valences verbales forment un ensemble plutôt qu'une liste. Plus précisément, étant donné un SN à type monté SN1 de la forme C/C:[NPi] où *NPi* est un signe-valence, et un verbe V1 de la forme s:EnsVal où *EnsVal* est l'ensemble de ses valences, NP1 se combine avec V1 pour former V2 ssi NP1 unifie avec une valence de l'ensemble des valences EnsVal du verbe. V2 est comme V1 sauf que la valence NPi a été retirée d'EnsVal et que certains traits de V1 ont été instanciés par la règle. Par ailleurs, la règle exige que les traits relatifs à l'information immédiate à gauche (*ImF*) et la forme du verbe (*FormeV*) -qui constituent autant de conditions à l'application de la combinaison en avant- unifient avec les traits correspondants de l'argument (*ImF* et *FormeV*), les conditions mises à l'application de la combinaison en arrière étant purement et simplement ignorées dans ce cas. Enfin, la règle exige que les traits du signe résultant soient instanciés comme en (13), à savoir que la classe morphologique du résultat soit celle de l'argument (*ClasseA*), que l'information sur le dernier signe à avoir concaténé à droite (*ImB*) soit transmise inchangée depuis l'argument, et que ce soit la classe morphologique du foncteur qui devienne l'information immédiate à gauche du résultat (*ClasseF*). Les traits du résultat conditionnés par l'ordre de combinaison (e.g. *FormeV'*) sont hérités des *Traits résultantF* (cf. (12)). La percolation de ces traits est cruciale pour notre traitement de l'ordre, comme le montre l'exemple (14) où le signe associé à *le* concatène avec le signe associé à *regarde* pour former un nouveau signe *le regarde*:

(14) Dérivation de *le regarde*

```

le:
C/(C: {(np:_:[obj ...]:_) ...}
      :[verbe ...]
      :([i or lui, ind]  $\Rightarrow$  fc  $\Rightarrow$  [le, ind], [i, imp]  $\Rightarrow$  bc  $\Rightarrow$  [le, imp]))
: {}
:[le ...]
:-

regarde:
s: {(np:_:[obj ...]:_), (np:_:[nom ...]:_), (np:_:[mod ...]:_)}
:[verbe, (i: i), ind]
:-

```

le regarde:  
 s: {(np:\_[nom ...]:\_), (np:\_[mod ...]:\_) }  
 :[verbe, (le: i), ind]  
 :\_

Lorsque *le* concatène par la gauche, les conditions mises à l'application de FC exigent que l'argument i.e. le verbe porte la valeur *i* or *lui* comme information immédiate à gauche (*i* représentant l'état initial où rien n'est encore venu concaténer par la gauche). Lorsque *le* concatène par la droite, les conditions sur BC vérifient qu'aucun élément ne s'est encore placé à la droite du verbe et que le mode de ce dernier est impératif. Aussi les phrases de (15) sont-elles analysées correctement:

- (15) (a) Il le lui donne.  
 (b) \*Il lui le donne.  
 (c) Donne le lui.  
 (d) \*Donne lui le.

La règle de combinaison en arrière (BC) fonctionne exactement comme FC, mis à part le fait que l'argument doit se trouver à gauche du foncteur et que l'information pertinente dans le champ *Sensibilité aux Règles* est celle liée à *bc*. Enfin, la grammaire comporte une règle d'effacement destinée à éliminer les valences optionnelles à la fin de l'analyse, et à définir les conditions de grammaticalité d'une phrase. Aucune règle supplémentaire n'est nécessaire.

### 3.2. EXPRESSION DES VARIABLES SOUS-JACENTES A L'ORDRE DES MOTS

Au lieu de faire usage de valeurs primitives comme *pre* ou *post*, FG fait interagir les traits et les règles de combinaison pour rendre compte des contraintes de linéarité évoquées en (9). La contrainte (9a) est exprimée par le champ *Sensibilité aux Règles*. (9b) (forme du verbe) et (9c) (classe morphologique du dernier foncteur à avoir concaténé) sont exprimées par le fait que les traits d'un foncteur qui conditionnent la linéarité comprennent les traits *FormeV* et *ImF/ImB*. (9d) est exprimée par le fait que les conditions qui sont invariantes pour une classe donnée de constituants (clitiques, constituants interrogatifs, lexicaux) se trouvent énoncées dans le champ *Sensibilité aux Règles* de ces constituants, comme c'est le cas par exemple pour les transitions autorisées entre clitiques présentées en (14). Les conditions qui varient selon la classe du constituant et la nature de la valence consommée sont exprimées plutôt dans les Traits des valences verbales. Par exemple, un SN lexical peut être sujet ou objet. S'il est sujet et qu'il se trouve à la gauche du verbe, il ne doit pas y avoir de constituant interrogatif entre lui et le verbe. S'il est sujet et qu'il se place à la droite du verbe, il doit lui être immédiatement adjacent. Une valence verbale est de la forme:

- (16) (np:\_[... X, Y ...]:\_)

où X et Y (soit variables anonymes, soit constantes) contraignent, entre autres, les valeurs possibles d'*ImF* et *ImB*. Pour mémoire, une valence est un signe faisant partie de l'ensemble des *Valences* d'un signe verbal. Le signe actif d'un SN au type monté est de la forme:

- (17) C: {(np:\_[... V1, V2 ...]:\_) ...}  
 :[verbe ...]  
 :([V1 ...] ⇒ fc ⇒ Z, [V2 ...] ⇒ bc ⇒ W)

Par règle, V1 et V2 du champ *Valences* de (17) doivent unifier respectivement avec X et Y de la valence verbale (16). Etant reprises ailleurs dans le signe, ces variables transmettent l'information aux Conditions sur les règles FC et BC. De plus, V1 et V2 du

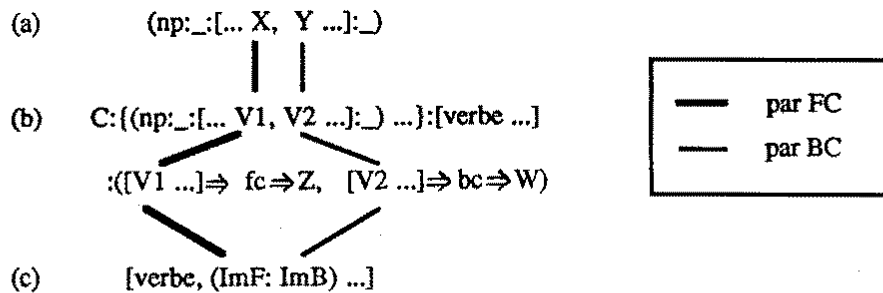


champ *Sensibilité aux Règles* du foncteur doivent unifier, par règle, avec certains traits particuliers des *Traits* du verbe, représentés en (18):

(18) [verbe, (ImF: ImB) ...]

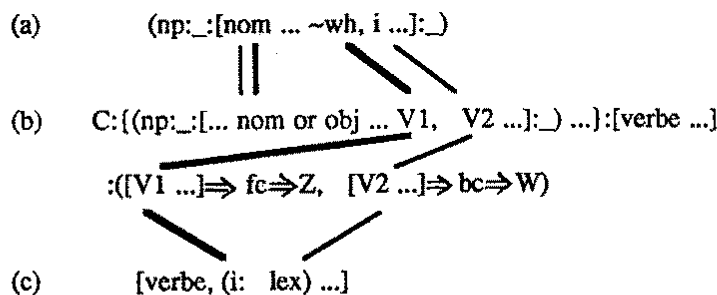
Le schéma (19) rend compte graphiquement du flux de l'information entre (16), (17) et (18); (19a), (19b) et (19c) correspondent respectivement à (16), (17) et (18). (19a) et (19c), qui représentent les champs *Valences* et *Traits* du même signe verbal, sont dissociés pour plus de clarté.

(19) Flux de l'information entre foncteur et argument



Supposons par exemple que nous ayons la valence nominative (20a), dans le signe verbal *téléphone à la fille*, dont les *Traits* sont comme en (20c), et le signe lexical *Jean* (20b):

(20) Flux de l'information entre *Jean* et *téléphone à la fille*



La concaténation par FC est autorisée (~wh étant compatible avec i), puisque la contrainte extraite de la valence verbale exige que le dernier constituant à avoir concaténé à gauche ne soit pas interrogatif. Mais la concaténation par BC échoue (i n'unifiant pas avec lex). Les exemples de (21) sont par conséquent analysés correctement (cf. Appendice):

- (21) (a) Jean téléphone à la fille.  
 (b) \*Téléphone à la fille Jean.  
 (c) \*Jean à quelle fille téléphone ?  
 (d) A quelle fille téléphone Jean ?

### 3.3. PROCEDURE DE COMBINAISON

Dans notre présentation des règles (13), il apparaît qu'il n'y a pas d'unification directe (i.e. de termes Prolog) entre l'ensemble des valences du signe actif du foncteur et l'ensemble des valences de l'argument. La procédure de combinaison est précisément la procédure qui permet de traiter les relations entre ensembles de valences. Sur un plan général, elle correspond à la formulation de (22):

$$(22) \quad (V_A - V_{SAF}) \cup V_F = V_R$$

où  $V_R$  est l'ensemble des valences du résultat  
 $V_A$  est l'ensemble des valences de l'argument  
 $V_{SAF}$  est l'ensemble des valences du signe actif du foncteur  
 $V_F$  est l'ensemble des valences du foncteur

Il est facile de vérifier que la dérivation de *le regarde* en (14) correspond à l'opération suivante:<sup>5</sup>

$$(23) \quad (\{obj, nom, mod\} - \{obj\}) \cup \{\} = \{nom, mod\}$$

Autrement dit, la concaténation des deux signes implique ici la *consommation* d'une valence de l'argument. Dans certains cas, aucune valence ne se trouve consommée, comme par exemple lors de la concaténation d'un verbe argument avec la particule négative *ne* dont nous schématisons le signe en (24):

$$(24) \quad \begin{array}{l} ne: \\ C/(C:\{\} \\ \quad :[verbe \dots] \\ \quad : \dots \\ \quad : \{\} \\ \quad :[ne \dots] \\ \quad :\_ \end{array}$$

La dérivation de *ne regarde* correspond à (25):

$$(25) \quad (\{obj, nom, mod\} - \{\}) \cup \{\} = \{obj, nom, mod\}$$

où l'ensemble des valences du verbe demeure inchangé. Un cas différent est celui d'un réflexif comme *se*, lequel doit consommer une valence objet et s'accorder en genre et en nombre avec le nominatif:

$$(26) \quad \begin{array}{l} se: \\ C/(C:\{(nom, G:N:p3), (obj, G:N:p3)\} \\ \quad :[verbe \dots] \\ \quad : \dots \\ \quad : \{(nom, G:N:p3)\} \\ \quad :[ne \dots] \\ \quad :\_ \end{array}$$

La dérivation de *se regarde* correspond à (27):

$$(27) \quad (\{obj, nom, mod\} - \{(obj, G:N:p3), (nom, G:N:p3)\}) \cup \{(nom, G:N:p3)\} \\ = \{(nom, G:N:p3), mod\}$$

où l'opération invoquée est la consommation de deux valences. La procédure unifie respectivement ces valences distinguées avec les valences *obj* et *nom* de l'ensemble des valences du verbe argument. Ces valences sont ensuite retirées de l'ensemble des valences du verbe, mais la liste des valences du foncteur réinsère la même valence *nom* instanciée dans l'ensemble des valences du résultat. La consommation puis la réinsertion de la valence *nom* a donc pour effet ici d'unifier les valeurs d'accord entre les valences *obj* et *nom* du verbe.

<sup>5</sup> Dans le reste de cette section, nous ne mentionnons que les informations pertinentes relativement aux valences. Un ensemble comme  $\{obj, nom, mod\}$  est par conséquent à interpréter comme un raccourci pour  $\{(np\_:[obj \dots]:\_), (np\_:[nom \dots]:\_), (np\_:[mod \dots]:\_)\}$ . Signalons qu'en FG, une valence optionnelle *mod* (modifieur) est systématiquement introduite dans l'ensemble des valences du verbe.

Mais les deux opérateurs génériques – et  $\cup$  de (22) ont un fonctionnement différent selon le type des objets auxquels ils s'appliquent. Le polymorphisme de ces opérateurs s'appuie sur une typologie du type (28) et sur des définitions comme celles de (29), où nous notons par les symboles –' et  $\cup'$  respectivement les opérations de réunion et de soustraction distinctes des opérations ensemblistes usuelles.

(28) Typologie des ensembles de valences

Un ensemble de valences V est de type

T\* ssi  $V = \{x, *y, Z\}$  où x est une valence distinguée et où Z est une variable ou un ensemble de valences

Tench ssi V est l'ensemble des valences d'une enchâssée (avant enchâssement)

Tile ssi  $\exists v \in V : v = \text{file}$

...

- (29) (a) Si  $B = \{x, *y, Z\}$  (i.e. est de type T\*)  
alors  $A -' B = (A - x)$  ssi  $\forall v \in (A - x)$ , v unifie avec y  
sinon  $A -' B = \perp$ .
- (b) Si A est de type Tench et si  $B = \{x, *y, Z\}$  (i.e. est de type T\*)  
alors  $A \cup' B = (A \cup Z)$  ssi  $\exists v \in A$  tel que w unifie avec x, et  
 $\forall v \in A$ ,  $v \neq w$  et v unifie avec y  
sinon  $A \cup' B = \perp$ .

...

Informellement, le symbole \* permet de procéder, en même temps qu'à la consommation, à une *vérification* sur un ensemble de valences. Autrement dit, la concaténation de deux signes implique alors la vérification d'une ou de plusieurs contraintes (et, le cas échéant, l'instanciation des valeurs correspondantes), l'ensemble résultant se trouvant formé par l'union des deux ensembles de départ plus les valeurs instanciées. Soit le signe simplifié pour le complémenteur *que*:

- (30) *que*:  
C/(C: {(s: {npX, ~nom, wh} ...) ...} ...)  
/(s: {(npX, ~nom, wh), (\*np, wh, op)} ...)  
: {(npX, ~nom, wh)}  
:[comp ...]  
:\_

La concaténation de *que* avec *regarde* applique (29a), comme le montre (31):

- (31)  $(V_A -_{T^*} V_{SAF}) \cup V_F = V_R$
- (a)  $(\{obj, mod\} - \{obj, wh\}) \cup \{obj, wh\} = \{obj, wh, (mod, wh, op)\}$  <sup>6</sup>
- (b)  $(\{mod\} - \{mod, wh\}) \cup \{mod, wh\} = \{mod, wh\}$

Pour concaténer *que* et *regarde*, la procédure de combinaison vérifie que toutes les valences du verbe (autre que celle qui est consommée) sont unifiaibles avec la valence \*np, ce qui -du point de vue linguistique- signifie que l'on peut extraire d'une phrase enchâssée finie une valence optionnelle ou bien non optionnelle (mais au plus *une* valence non optionnelle), et que l'extraction a notamment pour effet de marquer les valences restantes du verbe argument comme interrogatives (wh), et de vérifier que toutes ses valences obligatoires ont déjà été incorporées. Cela n'est possible que pour la

<sup>6</sup> Ce sont des principes indépendants de la grammaire qui empêchent la juxtaposition de deux (ou plus de deux) éléments interrogatifs déplacés en tête de phrase.

concaténation de *que* avec *SNnom regarde* ((31a)) et la concaténation de *que* avec *SNnom regarde SObj* ((31b)). Ce sont ces blocs qui fonctionnent ensuite comme foncteurs par rapport au verbe matrice, en y consommant la valence (s:{np, ~nom}...), e.g. le verbe *promet* dont nous présentons en (32) le signe simplifié:

- (32) promet:  
 s: {(np: \_:[à ...]:\_), (np: \_:[nom ...]:\_), (np: \_:[mod ...]:\_), (s: {np, ~nom}: [obj ...]:\_)}  
 : [verbe ...]  
 : \_

Lorsque le complémenteur foncteur *que* concatène avec une phrase comme *Jean regarde*, la spécification de *\*np* se trouve exprimée dans le foncteur et concerne aussi bien la liste des valences de n'importe quel verbe argument, ce qui veut dire qu'elle est valable pour l'enchâssement des phrases finies en général. Inversement, ce qui se passe entre un prédicat principal et une infinitive foncteur est déterminé par le prédicat principal, qui conditionne ainsi la liste des valences de l'infinitive foncteur, comme en témoignent les données liées à l'extraction. En règle générale, l'extraction se produit obligatoirement hors de l'infinitive (e.g. (33a)) mais elle est obligatoirement locale dans le cas des interrogatives indirectes (e.g. (33b)):

- (33) (a) A qui Jean dit d'envoyer un livre ?  
           \*Jean dit d'à qui envoyer un livre ?  
       (b) Jean cherche à qui envoyer un livre  
           \*A qui Jean cherche envoyer un livre ?

Dans le cas des constructions adjectivales à 'montée de l'objet', l'extraction se produit obligatoirement vers la matrice (elle ne peut être ni illimitée ni locale), et l'élément extrait peut être un item lexical:

- (34) Quel camion est difficile à charger ?  
       Le camion est difficile à charger

Enfin, dans le cas de *faire* et *laisser*, il peut y avoir extraction d'un clitique vers la matrice:

- (35) Jean le fait charger  
       Jean le laisse charger

Le traitement à réserver aux valences de l'infinitive est donc sujet à variations, et cette variabilité dépend de la nature du prédicat principal, en conséquence de quoi la spécification de *\*np* doit s'effectuer dans le prédicat principal, et la combinaison entre infinitive enchâssée et matrice obéir à (29b).

Linguistiquement parlant, on voit donc que l'union pure et simple entre les ensembles de valences du foncteur et de l'argument (moins la valence distinguée) est trop permissive; elle signifierait que les valences provenant d'ensembles distincts sont toujours susceptibles de se mélanger, ce qui est faux. Il est évident que certains ensembles de valences forment des domaines clos, que nous avons appelés *îles*. Ainsi, la typologie (28) intervient également pour les ensembles de valences des noms et des adjectifs, pour lesquels on a notamment les règles suivantes:

- (36) (a) Si A n'est ni de type Tench ni de type Tile, et  
           si B est de type Tile et n'est pas de type T\*  
           alors  $A \cup B = A$ .  
       (b) Si A est de type Tile et si B est de type Tile  
           alors  $A \cup B = A \cup B$ .  
       ...

La dérivation d'un énoncé comme *La fille triste de partir regarde Jean* met en relation l'ensemble des valences du nom *fille* et celui du prédicat adjectival *triste*, à savoir {île, mod} et {(np, wh), (\*np, wh, op), île, mod, (s:{np})} respectivement. Via (36b), l'ensemble des valences du signe *fille triste* est {(np, wh), (\*np, wh, op), île, mod, (s:{np})} i.e. l'union simple entre les ensembles de départ. L'infinitive *de partir* concatène avec *fille triste* via la règle (29b) en consommant la valence (s:{np}), ce qui laisse {île, mod} comme ensemble de valences pour *fille triste de partir*. Il est important de noter que le fait que l'ensemble des valences du déterminant *la* soit vide a pour effet (via (36a)) de ramener l'ensemble des valences du signe *la fille triste de partir* à l'ensemble vide. Cela signifie également qu'il est impossible de consommer la valence infinitive enchâssée une fois que l'on a formé le signe *la fille triste*, puisque celle-ci a été supprimée durant le processus de concaténation. On évite donc des phrases comme (37):

(37) \*La fille triste regarde Jean de partir

où l'on constate que les valences d'une île doivent former bloc avec le signe dans lequel elles apparaissent, et ne peuvent pas en général être transportées plus loin pour être consommées lors de concaténations ultérieures (cf. Baschung (1990) pour un traitement de ces problèmes).

#### 4. IMPLEMENTATION

Le formalisme UCG et son environnement informatique ont été développés au *Centre for Cognitive Science* de l'Université d'Edimbourg par Calder et al. (1986). Le logiciel PIMPLE comporte une implémentation Prolog de l'environnement linguistique PATR-II, ainsi qu'un analyseur de type 'shift-reduce'. Les extensions au logiciel initial liées à notre implémentation d'une grammaire du français ont été réalisées à l'Université Blaise Pascal Clermont II par Claire Gardent et Karine Baschung. Le système tourne sur SUN sous UNIX, et a subi une batterie importante de tests.

#### 5. CONCLUSION ET PERSPECTIVES

La grammaire FG traite les phénomènes centraux de la linéarité en français, i.e. les phrases déclaratives, interrogatives et négatives à tous les modes, avec des formes verbales simples ou composées: questions en oui/non, constituants interrogatifs inversion interrogative, placement des clitiques (y compris les réflexifs), auxiliaires (y compris l'accord entre l'objet antéposé et le participe), passives, enchâssées finies et infinitives (y compris certaines relatives), négation de phrases ou de constituants. Les dépendances à distance illimitée sont traitées lexicalement, sans qu'il soit besoin de recourir à des techniques comme le *gap threading* d'UCG ou la composition fonctionnelle de CCG (*Combinatory Categorical Grammar*, cf. Steedman (1986)). Enfin, des contraintes d'ordre non locales comme celles relatives à la distribution des particules négatives ou encore l'exigence pour une phrase dont le sujet est lexical et à droite du verbe de comporter un constituant interrogatif à la gauche du verbe (cf. (21d)) sont incorporées sans difficulté.

De même, il est possible d'élargir la couverture de la grammaire aux verbes supports qui posent des problèmes spécifiques et notoirement complexes pour les grammaires catégorielles, où ne procède que par concaténation d'éléments adjacents. Dans une construction à verbe support, le rôle de prédicat sémantique est tenu par le substantif (*Npred*) alors que le verbe est assimilable à un auxiliaire introduisant des valeurs aspectuelles:

(38) Cette nouvelle a un grand impact sur le public

Cette nouvelle prend un grand impact sur le public  
Cette nouvelle perd tout impact sur le public

En (38), on voit que c'est le Npred *impact* et non le verbe *a, prend* ou *perd* qui apporte ses arguments à la phrase; il est donc porteur de valences. Par ailleurs, la présence de certains compléments en excluant d'autres, la grammaire doit pouvoir exprimer le fait qu'une valence A d'un signe X interdit la consommation d'une valence B d'un signe Y, comme en (39) où la valence à Marie du verbe support *porte* interdit la valence pour Paul du Npred *admiration*:

(39) Pierre porte de l'admiration à Marie (\*pour Paul)

Cette interdépendance des compléments peut être respectée par la procédure de combinaison grâce à une vérification sur les ensembles de valences puis l'intervention de la règle d'effacement en fin d'analyse. Il faut également rendre compte de constructions avec permutation des compléments, où un constituant se trouve consommé sans avoir été requis, e.g. (40) où *éprouve* concatène avec *pour Marie* sans être sous-catégorisé pour ce complément:

(40) Pierre éprouve pour Marie une grande admiration

Les opérations ensemblistes effectuées par la procédure de combinaison rendent tout à fait possible un traitement de ces problèmes (cf. Emorine (1990)).

C'est pourquoi il apparaît qu'une grammaire lexicaliste comme FG constitue une synthèse prometteuse entre les aspects les plus intéressants des grammaires syntagmatiques d'une part et des grammaires catégorielles de l'autre. Comme en GPSG, dans notre formalisme les valences verbales ne sont pas intrinsèquement ordonnées, sans pour autant qu'il soit fait usage de stipulations sur la précédence linéaire (*LP-statements*). Ce dernier point est important dans la mesure où l'ordre des clitiqes en français fait intervenir des contraintes opérant sur des arbres locaux différents, ce qui est un problème pour l'approche GPSG (cf. Bès (1988)). A la différence de GPSG encore (et comme en UCG) il n'est pas besoin en FG de principes spécifiques d'instanciation des traits, puisque la circulation de l'information est assurée par les règles via le mécanisme d'unification.

A la différence d'UCG, remarquons que le nombre des règles de la grammaire est maintenu à trois (deux règles de combinaison plus une règle d'effacement). Quant à la procédure de combinaison, elle est définie par un nombre fini d'opérations entre ensembles de valences. Une conséquence évidente de la représentation des valences comme des ensembles et non comme des listes est cependant que l'efficacité de l'analyseur est fortement remise en cause: étant donné trois signes adjacents SN1 V SN2 par exemple, il y a deux moyens de combiner chaque SN avec le verbe et par conséquent deux dérivations; dans une phrase plus complexe le problème bien connu des fausses ambiguïtés (*spurious ambiguity*) se multiplie rapidement. On trouve dans Gardent et al. (1989) l'exposé d'une heuristique incorporée à notre analyseur, laquelle garantit l'unicité des analyses dans tous les cas où l'ambiguïté n'est pas réelle. Par ailleurs, la grammaire ne surgénère pas i.e. elle accepte toutes et seulement les phrases grammaticales. Ce résultat est dû au système de traits qui restreint les combinaisons possibles, et à l'élimination systématique de l'ambiguïté lexicale injustifiée - à savoir l'ambiguïté non motivée par des différences catégorielles ou sémantiques, et principalement destinée à permettre l'insertion d'un mot dans toutes ses positions autorisées dans une phrase. Enfin, la procédure de combinaison permet d'effectuer des disjonctions entre sous-ensembles de valences à l'intérieur de l'ensemble des valences d'un même signe, ce qui constitue une approche unifiante de la sous-catégorisation (cf. Baschung (1990)). On aura par exemple une seule entrée pour *dire' SN à SN, dire' que P à SN, dire' Inf à SN* et une entrée pour *dire" que Psubj à SN, dire" de Inf à SN*, correspondant à la sémantique *dire* vs. *ordonner* respectivement. Dans chacune de ces entrées, les

alternatives sont représentées par une même formule sémantique, ce qui rejoint des hypothèses émises récemment dans le cadre de la sémantique lexicale (cf. Boguraev et al. (1990) ou Briscoe et al. (1990)).

## REFERENCES

- Bar-Hillel, Y. (1953) A quasi-arithmétique notation for syntactic description, *Language*, 29, 47-58.
- Baschung, K. (1990) Grammaires d'unification à traits et contrôle des infinitives, Thèse de Doctorat, Université Blaise Pascal Clermont II. A paraître aux Editions ADOSA, Clermont-Ferrand.
- Baschung, K., Bès, G.G., Corluy, A. & Guillotin, T. (1987) Auxiliaries and Clitics in French UCG Grammar, *Proceedings of the Third European Chapter of the Association for Computational Linguistics*, Copenhagen, 173-178.
- Bès, G.G. (1988) Clitiques et constructions topicalisées dans une grammaire GPSG du français, *Lexique*, 6, 55-81.
- Bès, G.G. & Gardent, C. (1989) French Order without Order, *Proceedings of the Fourth European Chapter of the Association for Computational Linguistics*, Manchester, 249-255.
- Boguraev, B. & Pustejovsky, J. (1990) Lexical Ambiguity and Knowledge Representation, *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki.
- Briscoe, T., Copestake, A. & Boguraev, B. (1990) Lexical Semantics via Lexicology, *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki.
- Calder, J., Moens, M. & Zeevat, H. (1986) A UCG Interpreter, ESPRIT Project 393 ACORD, Deliverable T2.6, Centre for Cognitive Science, University of Edinburgh.
- Emorine, M. (1990) Les verbes supports dans une grammaire catégorielle du français, communication présentée au colloque *L'ordre des mots dans les grammaires catégorielles*, Projet ESPRIT Basic Research Action 3175 DYANA, 25-27 mai 1990, Clermont-Ferrand.
- Gardent, C., Bès, G.G., Jurie, P.F. & Baschung, K. (1989) Efficient Parsing for French, *Proceedings of the Conference of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, 280-287.
- Karttunen, L. (1986) Radical Lexicalism, Report n° CSLI-86-68, Center for the Study of Language and Information. Paper presented at the Conference on Alternative Conceptions of Phrase Structure, July 1986, New-York.
- Steedman, M.J. (1986) Incremental Interpretation in Dialogue, ESPRIT Project 393 ACORD, Deliverable T2.4, Centre for Cognitive Science, University of Edinburgh.
- Uszkoreit, H. (1987) Word Order and Constituent Structure in German, *CSLI Lectures Notes* n° 8, Stanford.
- Zeevat, H., Klein, E. & Calder, J. (1987) An Introduction to Unification Categorical Grammar, in Haddock, N.J., Keim, E. & Morrill, G. (eds.) *Edinburgh Working Papers in Cognitive Science*, vol.1: *Categorical Grammar, Unification Grammar, and Parsing*.

## APPENDICE - Contraintes sur l'ordre

La matrice ci-dessous représente les traits des valences nominatives (a) ou non-nominatives (b) des signes verbaux (i.e. correspondant à (20a)), et les traits dans les valences des signes actifs des SN lexicaux (c) ou interrogatifs (d) (cf. (20b)). Les colonnes représentent des traits spécifiques des deux types de valences (cf. (11)).

	ImF	ImB	ImF	ImB
(a) valence nominative	~wh	i	~wh	k
(b) valence non nominative	k	—	~wh	—
(c) SN lexical	V1	$\bar{V}2$	$\bar{V}1$	$\bar{V}2$
(d) SN interrogatif	—	—	$\bar{V}1$	$\bar{V}2$

La matrice exprime les contraintes suivantes (le constituant qui introduit la contrainte se trouve en italiques):

- (a) Un SN lexical sujet à la gauche d'un verbe ne peut pas être immédiatement suivi par un constituant interrogatif, e.g. \**Jean* quel homme regarde ? (avec *Jean* sujet)
- (b) Un SN lexical sujet à la droite d'un verbe doit lui être immédiatement adjacent, e.g. \*Quel cadeau présente à *Marie Pierre* ?
- (c) Un SN sujet interrogatif à gauche du verbe ne peut pas être immédiatement suivi par un constituant interrogatif, e.g. \**Qui* quel homme regarde ?
- (d) Il ne peut pas y avoir de SN interrogatif sujet à la droite du verbe, e.g. \*Jean regarde *qui* ? (avec *qui* sujet)
- (e) Les SN lexicaux non-nominatifs ne peuvent être placés à la gauche du verbe, e.g. \**Marie Pierre* regarde
- (f) Un SN interrogatif non-nominatif à la gauche du verbe ne peut pas être immédiatement suivi par un constituant interrogatif, e.g. \**Qui* qui regarde ?